

# OpenVAS Vulnerability Assessment and Management Project

## Table of Contents

1. Project Overview
2. What is OpenVAS?
3. Prerequisites
4. Installation Guide
5. Configuration and Setup
6. Vulnerability Scanning Process
7. Results Analysis
8. Critical Information to Look For
9. Understanding Vulnerabilities, Versions, and Services
10. Security Best Practices
11. Framework Compliance
12. Advantages and Disadvantages
13. Recommendations
14. Conclusion
15. References

## Project Overview

**Project Name:** OpenVAS Vulnerability Assessment and Security Audit

**Purpose:** This project demonstrates the complete implementation of OpenVAS (Open Vulnerability Assessment System) for network vulnerability scanning, assessment, and remediation planning in a controlled virtual environment.

**Environment:** Kali Linux running on Oracle VirtualBox

**Scope:** Local network vulnerability assessment, security posture evaluation, and compliance verification

## What is OpenVAS?

### Definition

OpenVAS (Open Vulnerability Assessment Scanner) is a comprehensive open-source vulnerability scanning and management solution. It is part of the Greenbone Vulnerability Management (GVM) framework and provides capabilities for:

- Network vulnerability detection
- Security configuration assessment
- Compliance auditing
- Continuous security monitoring
- Automated vulnerability testing

### Key Components

1. **GVM Scanner (OpenVAS)** - The actual scanning engine
2. **GVM Manager (gvmd)** - Central management daemon
3. **Greenbone Security Assistant (GSA)** - Web-based user interface
4. **PostgreSQL Database** - Stores scan results and configurations
5. **Redis** - Cache for NVT (Network Vulnerability Tests) data

### How OpenVAS Works

OpenVAS operates by:

1. Performing unauthenticated and authenticated network scans
2. Testing for over 50,000 known vulnerabilities (NVTs)
3. Identifying security misconfigurations
4. Detecting outdated software versions
5. Providing severity ratings and remediation guidance

## **Prerequisites**

### **Hardware Requirements**

- **RAM:** Minimum 4GB (8GB recommended)
- **Disk Space:** Minimum 20GB free space
- **Processor:** 2+ cores recommended
- **Network:** Active network connection

### **Software Requirements**

- Oracle VirtualBox (Latest version)
- Kali Linux ISO (Latest version)
- At least one target machine for scanning (can be another VM)

### **Knowledge Prerequisites**

- Basic Linux command line operations
- Understanding of networking concepts (IP addresses, ports, protocols)
- Basic cybersecurity concepts
- Virtual machine management

## **Installation Guide**

### **Step 1: Set Up Kali Linux on VirtualBox**

1. Download Kali Linux ISO from official website
2. Create new VM in VirtualBox:
  - Name: Kali-OpenVAS
  - Type: Linux
  - Version: Debian (64-bit)
  - RAM: 4096 MB minimum
  - Storage: 40GB dynamically allocated

3. Configure VM Settings:
  - Network: NAT or Bridged Adapter
  - Enable PAE/NX
  - Allocate 2+ CPU cores
4. Install Kali Linux following the standard installation process

## Step 2: Update Kali Linux System

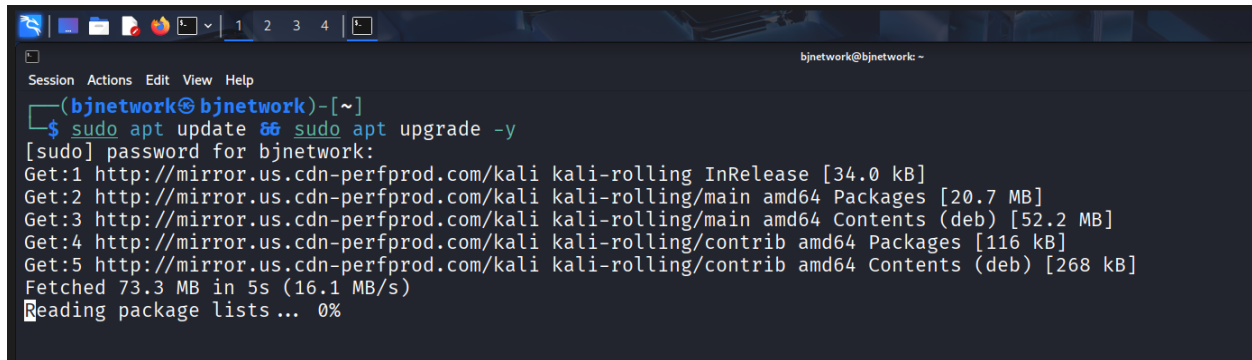
Open terminal and run:

bash

sudo apt update && sudo apt upgrade -y

sudo apt dist-upgrade -y

sudo reboot



```
bjnetwork@bjnetwork: ~  
$ sudo apt update && sudo apt upgrade -y  
[sudo] password for bjohnson:  
Get:1 http://mirror.us.cdn-perfprod.com/kali kali-rolling InRelease [34.0 kB]  
Get:2 http://mirror.us.cdn-perfprod.com/kali kali-rolling/main amd64 Packages [20.7 MB]  
Get:3 http://mirror.us.cdn-perfprod.com/kali kali-rolling/main amd64 Contents (deb) [52.2 MB]  
Get:4 http://mirror.us.cdn-perfprod.com/kali kali-rolling/contrib amd64 Packages [116 kB]  
Get:5 http://mirror.us.cdn-perfprod.com/kali kali-rolling/contrib amd64 Contents (deb) [268 kB]  
Fetched 73.3 MB in 5s (16.1 MB/s)  
Reading package lists... 0%
```

## Step 3: Install OpenVAS (GVM)

bash

*Install GVM (Greenbone Vulnerability Management)*

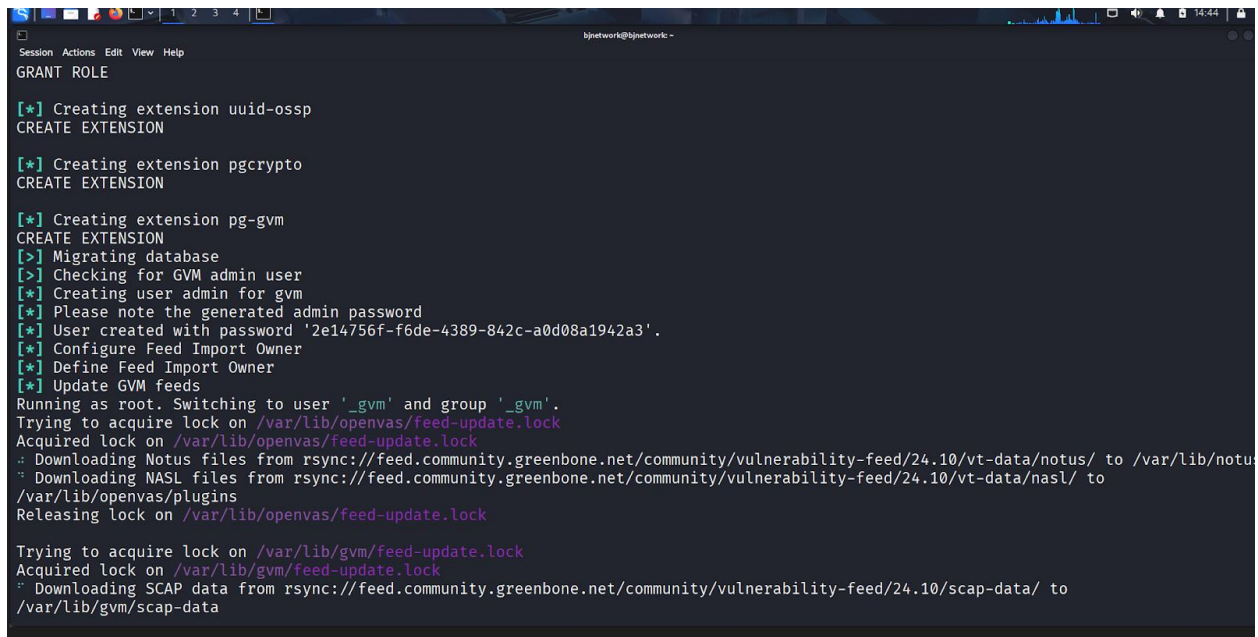
sudo apt install gvm -y

## Step 4: Run Initial Setup

bash

*Run the setup script*

sudo gvm-setup



```
Session Actions Edit View Help
GRANT ROLE

[*] Creating extension uuid-ossdp
CREATE EXTENSION

[*] Creating extension pgcrypto
CREATE EXTENSION

[*] Creating extension pg-gvm
CREATE EXTENSION
[>] Migrating database
[>] Checking for GVM admin user
[*] Creating user admin for gvm
[*] Please note the generated admin password
[*] User created with password '2e14756f-f6de-4389-842c-a0d08a1942a3'.
[*] Configure Feed Import Owner
[*] Define Feed Import Owner
[*] Update GVM feeds
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/openvas/feed-update.lock
Acquired lock on /var/lib/openvas/feed-update.lock
.: Downloading Notus files from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/vt-data/notus/ to /var/lib/notus
.: Downloading NASL files from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/vt-data/nasl/ to /var/lib/openvas/plugins
Releasing lock on /var/lib/openvas/feed-update.lock

Trying to acquire lock on /var/lib/gvm/feed-update.lock
Acquired lock on /var/lib/gvm/feed-update.lock
.: Downloading SCAP data from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/scap-data/ to /var/lib/gvm/scap-data
```

This process may take 15-30 minutes. The script will:

- Configure PostgreSQL database
- Download and update vulnerability feeds (NVTs)
- Create certificates
- Configure scanner and manager
- Set up the admin user

**IMPORTANT:** Save the admin password displayed at the end of setup!

## Step 5: Verify Installation

bash

*Check GVM services status*

sudo gvm-check-setup

Expected output should show all checks passed.

```
Session Actions Edit View Help
[*] Update GVM feeds
Running as root. Switching to user '_gvm' and group '_gvm'.
Trying to acquire lock on /var/lib/openvas/feed-update.lock
Acquired lock on /var/lib/openvas/feed-update.lock
: Downloading Notus files from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/vt-data/notus/ to /var/lib/notus
: Downloading NASL files from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/vt-data/nasl/ to /var/lib/openvas/plugins
Releasing lock on /var/lib/openvas/feed-update.lock

Trying to acquire lock on /var/lib/gvm/feed-update.lock
Acquired lock on /var/lib/gvm/feed-update.lock
: Downloading SCAP data from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/scap-data/ to /var/lib/gvm/scap-data
: Downloading CERT-Bund data from rsync://feed.community.greenbone.net/community/vulnerability-feed/24.10/cert-data/ to /var/lib/gvm/cert-data
: Downloading gvm data from rsync://feed.community.greenbone.net/community/data-feed/24.10/ to /var/lib/gvm/data-objects/gvmd
Releasing lock on /var/lib/gvm/feed-update.lock

[*] Checking Default scanner
08b69003-5fc2-4037-a479-93b440211c73 OpenVAS /run/ospd/ospd-openvas.sock 0 OpenVAS Default
[i] No need to alter default scanner
```

```
Session Actions Edit View Help
OK: CERT data found in /var/lib/gvm/cert-data.
Step 5: Checking Postgresql DB and user ...
OK: Postgresql version and default port are OK.
gvmd | _gvm | UTF8 | libc | en_US.UTF-8 | en_US.UTF-8 | | |
16453|pg-gvm|10|2200|f|22.6||
OK: At least one user exists.
Step 6: Checking Greenbone Security Assistant (GSA) ...
OK: Greenbone Security Assistant is present in version Deamon 24.12.2~git.
Step 7: Checking if GVM services are up and running ...
Starting gvmd service
Waiting for gvmd service
OK: gvmd service is active.
Starting gsad service
Waiting for gsad service
OK: gsad service is active.
Step 8: Checking few other requirements...
OK: nmap is present.
OK: ssh-keygen found, LSC credential generation for GNU/Linux targets is likely to work.
OK: nsis found, LSC credential package generation for Microsoft Windows targets is likely to work.
OK: xsltproc found.
WARNING: Your password policy is empty.
SUGGEST: Edit the /etc/gvm/pwppolicy.conf file to set a password policy.
Step 9: Checking greenbone-security-assistant...
OK: greenbone-security-assistant is installed

It seems like your GVM-25.04.0 installation is OK.
```

## Configuration and Setup

### Step 1: Start GVM Services

bash

*Start all GVM services*

*Verify services are running*

```
sudo systemctl status ospd-openvas
```

```
sudo systemctl status gvmd
```

```
sudo systemctl status gsad
```

```
(bjnetwork@bjnetwork)-[~]
$ sudo gvm-start
[i] GVM services are already running

(bjnetwork@bjnetwork)-[~]
$ sudo systemctl status gvm
● gvm.service - Greenbone Vulnerability Manager daemon (gvm)
   Loaded: loaded (/usr/lib/systemd/system/gvm.service; disabled; preset: disabled)
   Active: active (running) since Thu 2026-01-22 16:24:09 EST; 3min 28s ago
  Invocation: 4fa7165d1f7a4b438dd537937269d4bc
     Docs: man:gvm(8)
  Process: 12385 ExecStart=/usr/sbin/gvmd --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm (code=exited, status=0/SUCCESS)
 Main PID: 12387 (gvmd)
    Tasks: 4 (limit: 4657)
   Memory: 464.4M (peak: 465.5M)
      CPU: 1min 1.466s
   CGroup: /system.slice/gvmd.service
           └─12387 "gvmd: Waiting" --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm
           └─12406 gpg-agent --homedir /var/lib/gvm/gvmd/gnupg --use-standard-socket --daemon
           └─12423 "gvmd: Synchron" --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm
           └─12428 "gvmd: Syncing" --osp-vt-update=/run/ospd/ospd-openvas.sock --listen-group=_gvm

Jan 22 16:24:08 bjnetwork systemd[1]: Starting gvm.service - Greenbone Vulnerability Manager daemon (gvm)...
Jan 22 16:24:08 bjnetwork systemd[1]: gvm.service: Can't open PID file '/run/gvmd/gvmd.pid' (yet?) after start: No such file or directory
Jan 22 16:24:09 bjnetwork systemd[1]: Started gvm.service - Greenbone Vulnerability Manager daemon (gvm).

lines 1-19/19 (END)
```

[illegible]

## Step 2: Update Vulnerability Feeds

bash

*Update NVT feeds*

```
sudo greenbone-feed-sync --type GVMD_DATA
```

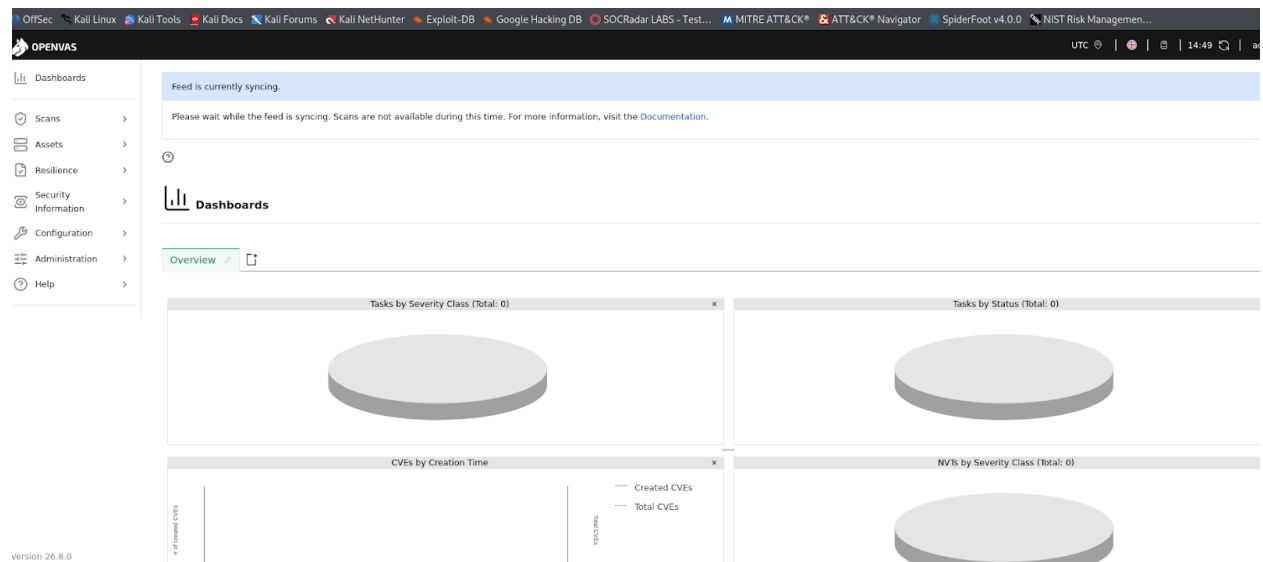
```
sudo greenbone-feed-sync --type SCAP
```

```
sudo greenbone-feed-sync --type CERT
```

*Alternative single command*

```
sudo runuser -u _gvm -- greenbone-nvt-sync
```

This may take 1-2 hours for first-time updates.



Type	Content	Origin	Version	Status
NVT	<a href="#">NVTs</a>	Greenbone Community Feed	20260121T0706	Current
SCAP	<a href="#">CVEs</a> <a href="#">CPEs</a>	Greenbone SCAP Data Feed	20260121T0510	Update
CERT	<a href="#">CERT-Bund Advisories</a> <a href="#">DFN-CERT Advisories</a>	Greenbone CERT Data Feed	20260121T0432	Update
GVMD_DATA	<a href="#">Compliance Policies</a> <a href="#">Port Lists</a> <a href="#">Report Formats</a> <a href="#">Scan Configs</a>	Greenbone Data Objects Feed	20260121T0511	Update



### Step 3: Access Web Interface

1. Open Firefox browser in Kali Linux
2. Navigate to: <https://127.0.0.1:9392> or <https://localhost:9392>
3. Accept the security certificate warning
4. Login with:
  - Username: admin
  - Password: (from gvm-setup output)

### Step 4: Create Additional User (Optional)

bash

*Create a new user*

```
sudo runuser -u _gvm -- gvmc --create-user=scanner_user --password=SecurePass123
```

*Modify existing user password*

```
sudo runuser -u _gvm -- gvmc --user=admin --new-password=NewSecurePass123
```

## Vulnerability Scanning Process

### Step 1: Define Target

**Navigate:** Configuration → Targets → New Target

#### Configuration:

- Name: Lab Network Scan
- Hosts: Manual entry or from file
  - Single IP: 192.168.1.100
  - Range: 192.168.1.1-254
  - CIDR: 192.168.1.0/24
- Port List: All IANA assigned TCP and UDP

**Screenshot Location:** screenshots/11\_target\_creation/

## Step 2: Configure Scan Task

**Navigate:** Scans → Tasks → New Task

### Configuration:

- Name: Network Security Assessment
- Scan Targets: Select created target
- Scanner: OpenVAS Default
- Scan Config: Choose based on scan type:
  - **Full and Fast:** Comprehensive, faster scan
  - **Full and Deep:** Most thorough, slower
  - **System Discovery:** Quick host discovery
  - **Full and Fast Ultimate:** Maximum coverage

### Advanced Options:

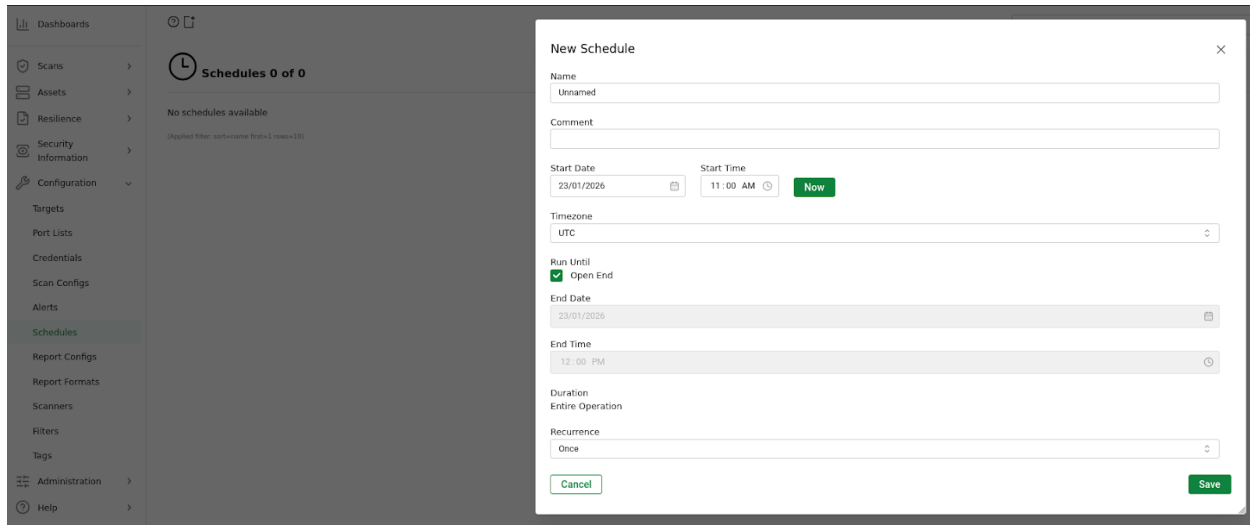
- Maximum Concurrency: 4-10 (adjust based on resources)
- Network Source Interface: Auto
- Order for target hosts: Sequential or Random

## Step 3: Create Scan Schedule (Optional)

**Navigate:** Configuration → Schedules → New Schedule

### Configuration:

- Name: Weekly Security Scan
- First Time: Select date/time
- Period: Weekly
- Duration: 2 hours



## Step 4: Launch Scan

1. Navigate to: Scans → Tasks
2. Select your task
3. Click the play button (▶) to start scan
4. Monitor progress in real-time

## Important Commands During Scanning

bash

*Monitor system resources*

htop

*Check scanner process*

ps aux | grep openvas

*View scanner logs*

sudo tail -f /var/log/gvm/openvas.log

*Check database activity*

```
sudo -u postgres psql gvm -c "SELECT COUNT(*) FROM results;"
```

0[|||||] 4.7% Tasks: 129, 478 thr, 85 kthr; 1 running

1[|||||] 5.6% Load average: 0.17 0.24 0.21

Mem[|||||] 12.06G/3.99G Uptime: 19:29:56

Swp[|||||] 696M/4.17G

Main		I/O									
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
54075	bjnetwork	20	0	10.5G	173M	70276	S	3.4	4.2	2h35:04	/usr/lib/firefox-esr/firefox-esr -contentproc -isForBrowser -prefsHandle 0:41050 -prefMapHandle 1:271586 -jsInitH
680	root	20	0	788M	202M	67260	S	2.7	5.0	4:51:99	/usr/lib/xorg/Xorg :0 -seat seat0 -auth /var/run/lightdm/root/:0 -nolisten tcp vt/ -novtswitch
54635	bjnetwork	20	0	10.5G	194M	0	S	2.0	4.8	0:17:70	/usr/lib/firefox-esr/firefox-esr -contentproc -isForBrowser -prefsHandle 0:41050 -prefMapHandle 1:271586 -jsInitH
54636	bjnetwork	20	0	10.5G	194M	0	S	1.4	4.8	0:07:51	/usr/lib/firefox-esr/firefox-esr -contentproc -isForBrowser -prefsHandle 0:41050 -prefMapHandle 1:271586 -jsInitH
1301	bjnetwork	20	0	559M	39368	18280	S	0.7	0.9	0:54:15	xfwm4
53628	bjnetwork	20	0	3235M	365M	0	S	0.7	8.9	0:21:71	/usr/lib/firefox-esr/firefox-esr
53723	bjnetwork	20	0	3235M	365M	0	S	0.7	8.9	6h18:45	/usr/lib/firefox-esr/firefox-esr
54109	bjnetwork	20	0	10.5G	194M	0	S	0.7	4.8	0:03:58	/usr/lib/firefox-esr/firefox-esr -contentproc -isForBrowser -prefsHandle 0:41050 -prefMapHandle 1:271586 -jsInitH
1	root	20	0	25408	12712	9376	S	0.0	0.3	0:02:39	/sbin/init splash

```
$ ps aux | grep openvas
redis 12172 0.1 4.5 435800 189420 ? Ssl Jan22 1:55 /usr/bin/redis-server unixsocket:/run/
_gvm 12205 0.1 4.8 524132 203464 ? Sl Jan22 1:52 /usr/bin/python3 /usr/bin/ospd-openvas
f
_gvm 12207 0.0 0.8 457488 36128 ? Sl Jan22 0:03 /usr/bin/python3 /usr/bin/ospd-openvas
f
_gvm 12387 0.0 6.9 588960 291160 ? SL Jan22 0:14 gvm: Waiting --osp-vt-update=/run/os
bjnetwo+ 175142 0.0 0.0 6748 2388 pts/0 S+ 10:11 0:00 grep --color=auto openvas
(bjnetwork@bjnetwork) [~]
```

Results Analysis

Step 1: View Scan Results

Navigate: Scans → Reports → Select completed scan



Report Dashboard Elements

1. Severity Distribution Graph

- High (Red): Critical vulnerabilities requiring immediate action
- Medium (Orange): Significant security issues
- Low (Yellow): Minor security concerns
- Log (Blue): Informational findings

## 2. Vulnerability Count Summary

- Total vulnerabilities found
- Breakdown by severity
- False positive indicators

## 3. Host Overview

- All scanned hosts
- Vulnerabilities per host
- Service detection results

## Step 2: Detailed Vulnerability Analysis

Click on any vulnerability to view:

### Key Information Fields:

#### 1. NVT Details

- NVT Name
- OID (Object Identifier)
- CVE ID (if applicable)
- CVSS Score

#### 2. Summary

- Clear description of vulnerability
- Affected component
- Attack vector

#### 3. Impact

- Confidentiality impact
- Integrity impact
- Availability impact

#### 4. **Solution**

- Remediation steps
- Patches or workarounds
- Configuration changes

#### 5. **References**

- CVE links
- Vendor advisories
- Security bulletins

### **Step 3: Filter and Sort Results**

#### **Filtering Options:**

- By Severity: High, Medium, Low, Log
- By Host: Specific IP addresses
- By Port: Specific services
- By NVT: Specific vulnerability types

bash

*Export results via CLI*

```
sudo -u _gvm gvm-cli socket --xml "<get_reports report_id='REPORT_ID'/>"
```

## **Critical Information to Look For**

### **1. High and Critical Vulnerabilities**

#### **What to Look For:**

- CVSS Score 7.0 or higher
- Remote Code Execution (RCE) vulnerabilities
- Authentication bypass issues
- SQL Injection vulnerabilities
- Cross-Site Scripting (XSS) in web applications
- Privilege escalation vulnerabilities

#### **Example Critical Findings:**

- EternalBlue (MS17-010)
- BlueKeep (CVE-2019-0708)
- Log4Shell (CVE-2021-44228)
- ProxyShell Exchange vulnerabilities

### **2. Missing Security Patches**

#### **What to Look For:**

- Outdated operating system versions
- Unpatched software applications
- End-of-Life (EOL) software
- Missing security updates

#### **How to Identify:**

- Check "Version Detection" results
- Compare against vendor security bulletins
- Review "Product Detection" findings